INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT

UDC 004.01

Original article

https://doi.org/10.23947/2687-1653-2022-22-4-384-390

Development of Architecture for Connecting a System Module for People with Disabilities

Alexey A. Baskakov ✉, **Alexey G. Tarasov**

Moscow State University of Technology (MSUT "STANKIN"), 1, Vadkovsky Lane, Moscow, Russian Federation

✉ aleks.baskakov@mail.ru

Abstract

Introduction. To develop new system modules of software to help employees with disabilities, it is required to work out an architectural solution for the interaction of all parts of the system. As a result of the analysis and design, it is necessary to obtain a software architecture that must meet a number of standard requirements. First of all, it should be safe. To do this, you should take into account the error logging system, event auditing, the possibility of disabling the functionality immediately after putting it into commercial operation, internal mechanisms for validating client input requests and server responses. This study is aimed at the development of basic system maintenance options, the analysis of exception cases under interacting with the user for further evaluation of the architecture efficiency, and the direct project development.

Materials and Methods. The architectural decision was carried out using the Unified Modeling Language (UML), which helps to build visual images of the life cycle and interaction of all components of the system. The syntax of the UML deployment diagram was used to study the interaction of the main modules of the future system, and the syntax of the UML sequence diagram was used to process the lifecycle. A use case diagram was also applied to describe the main use cases. To study the interaction of the main modules of the future system, the UML deployment diagram syntax was used. For life cycle processing, the UML sequence diagram syntax was applied. In addition, a use case diagram was applied to describe the base use cases.

Results. An architecture that has a scheme for the interaction of individual modules and systems, as well as options for using the software package for the future implementation of the software product, has been developed. The proposed system architecture meets the requirements of security, reliability (fault tolerance), and performance. The authors have fixed the functional requirements of the system of assistance to employees of enterprises with hearing problems for the possibility of their employment and work on the telecommunication Internet. Basic variations of system maintenance have been developed.

Discussion and Conclusions. Building a competent architecture provides taking into account cases that go beyond the normal use of the system, and applying a fuzzy model to determine the system efficiency. Further in-depth description of deployment and operation options will enable to implement an efficient and productive system.

Keywords: UML deployment diagram, UML sequence diagram, software architecture, commercial software, UML use case diagram.

Acknowledgements. The authors would like to thank Yuri G. Kogan, Cand.Sci. (Engineering), associate professor of the Department of Information Technologies and Computing Systems, STANKIN Moscow State Technical University, who participated in the development of the system use cases as a subject matter expert.

For citation. A. A. Baskakov, A. G. Tarasov. Development of Architecture for Connecting a System Module for People with Disabilities. Advanced Engineering Research, 2022, vol. 22, no. 4, pp. 384–390. https://doi.org/10.23947/2687-1653-2022-22-4-384-390

Introduction. Previously, the causes of disability among the population with disabilities were investigated through expert evaluation using the method of pairwise comparisons by Thomas L. Saati. As a result, it has been discovered that for improving the ability to work, it is required to develop a comprehensive system that would solve the problem of hearing loss and make it possible to work in remote support centers, where understanding the interlocutor is mandatory [1]. Before evaluating the efficiency, you need to work out all possible options for system maintenance. This will determine the service architecture efficiency associated with the performance of each module of the desired level of service. To reduce labor costs at the development stage, it is required to determine in advance exception cases when the module is working with transcription.

The scientific novelty of this study is the efficiency of the architecture in terms of tasks to be solved (service options).

The assessment was carried out using a fuzzy model of expert opinions in this field. It goes beyond the scope of discussion of this article.

To develop the architecture, you need to consider the following parameters:

– cross-platform — the software product should cover the main operating systems;

– fault tolerance — the system should work stably in case of interruptions in operation;

– security — the system must support logging, monitoring and auditing of user events;

– horizontal scalability — with an increase in the number of clients, the software product must work at the same speed by increasing the number of servers;

– performance — the system should work without delays in real time.

Planning and describing the software architecture is an important and necessary step before the direct development of a client-server application. The system should have a module for speech recognition and translate it into text, thereby allowing people with hearing problems to work on remote support.

Materials and Methods.

UML is a special modeling language that is used in the development of the architecture of computing systems, software, network architecture, and in the construction of business processes[1] [2, 3].

The following schemes were used to describe the project architecture:

1. Sequence diagram describes the process of receiving calls and processing them from the point of view of an employee of the organization[2].

2. Use case diagram describes the use cases of the system.

3. Deployment diagram describes the architecture of the system.

[1] Khammatova LA. Universal UML Modeling Language, Basic Diagrams and Usage Problems. In: ‘Proryvnye nauchnye issledovaniya: problemy, zakonomernosti, perspektivy”, Coll. of Papers XIII Int. Sci.-Pract. Conf. Penza, 2019. P. 88–90. (In Russ.)

[2] Teslenko IB, Tsarev AO. Osobennosti informatsionnogo proektirovaniya s ispol'zovaniem yazyka UML. In: “Innovatsionnoe razvitie sotsial'no-ehkonomicheskikh sistem: usloviya, rezul'taty i vozmozhnosti”, Proc. V Int. Sci.-Pract. Conf. Orekhovo-Zuyevo, 2017. P. 174–177. (In Russ.)

Research Results. The construction of UML diagrams made it possible to determine the system service quality criteria, to take into account exception cases and ways of processing them, and the construction of a deployment diagram — to determine the system efficiency using a fuzzy model.

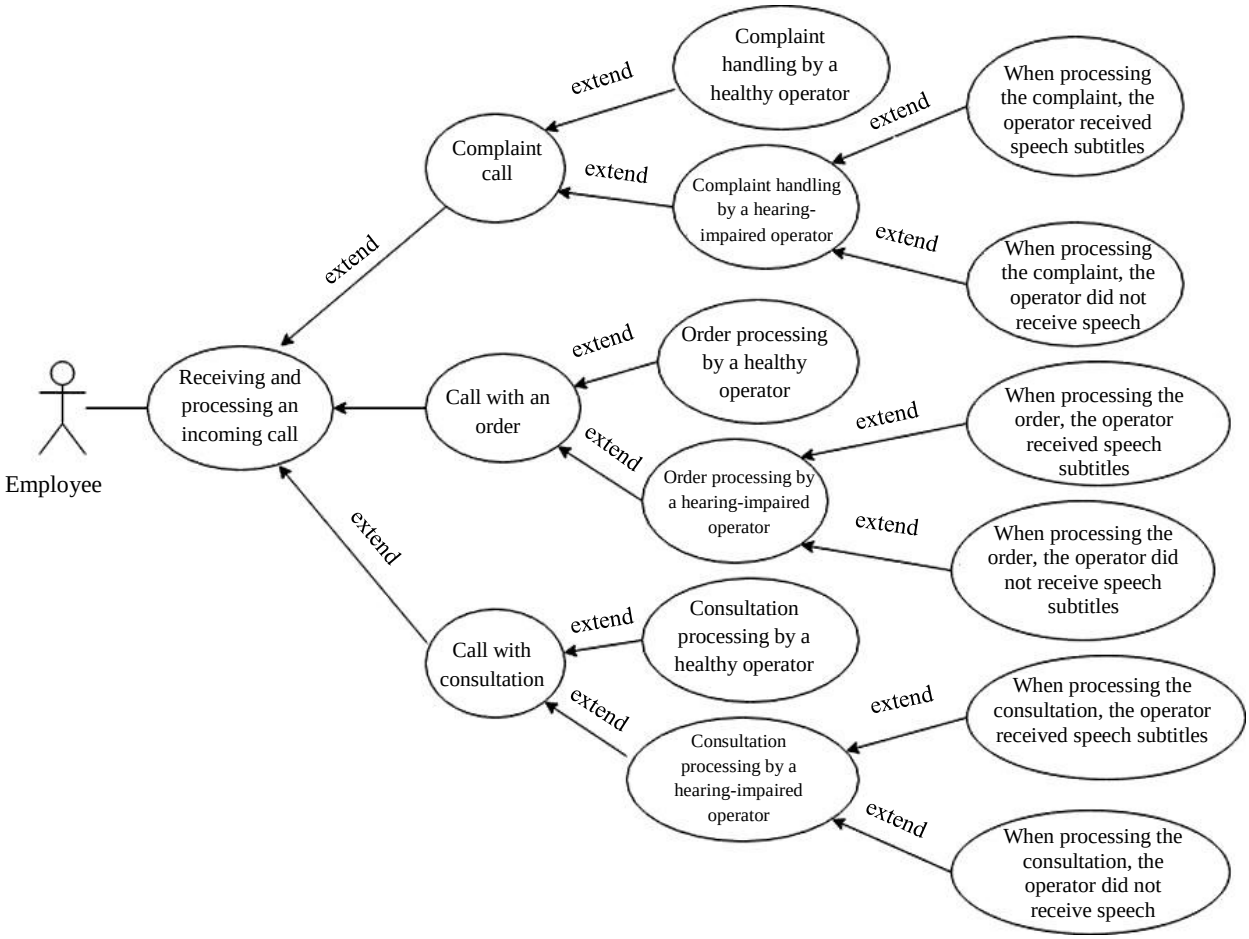

Fig. 1. Diagram of system use cases (the authors' figure)

Figure 1 shows a use case diagram developed by the authors of this article, which demonstrates that the main remote support requests come for the following reasons: complaints about the provision of services by the enterprise, consultation, and ordering services. At that, it should be borne in mind that the system of automatic generation of subtitles may work incorrectly for some reasons, i.e., it could not recognize speech, filtering of prohibited words was not passed, etc. In this case, you should switch the client to a robotic system or another operator.

The robotic system is an automatic voice generation module with its own life cycle, which enables to inform the client about technical problems during a call. The module also provides switching to another free operator at the will of the client or taking a queue in case of a heavy load at a given time.

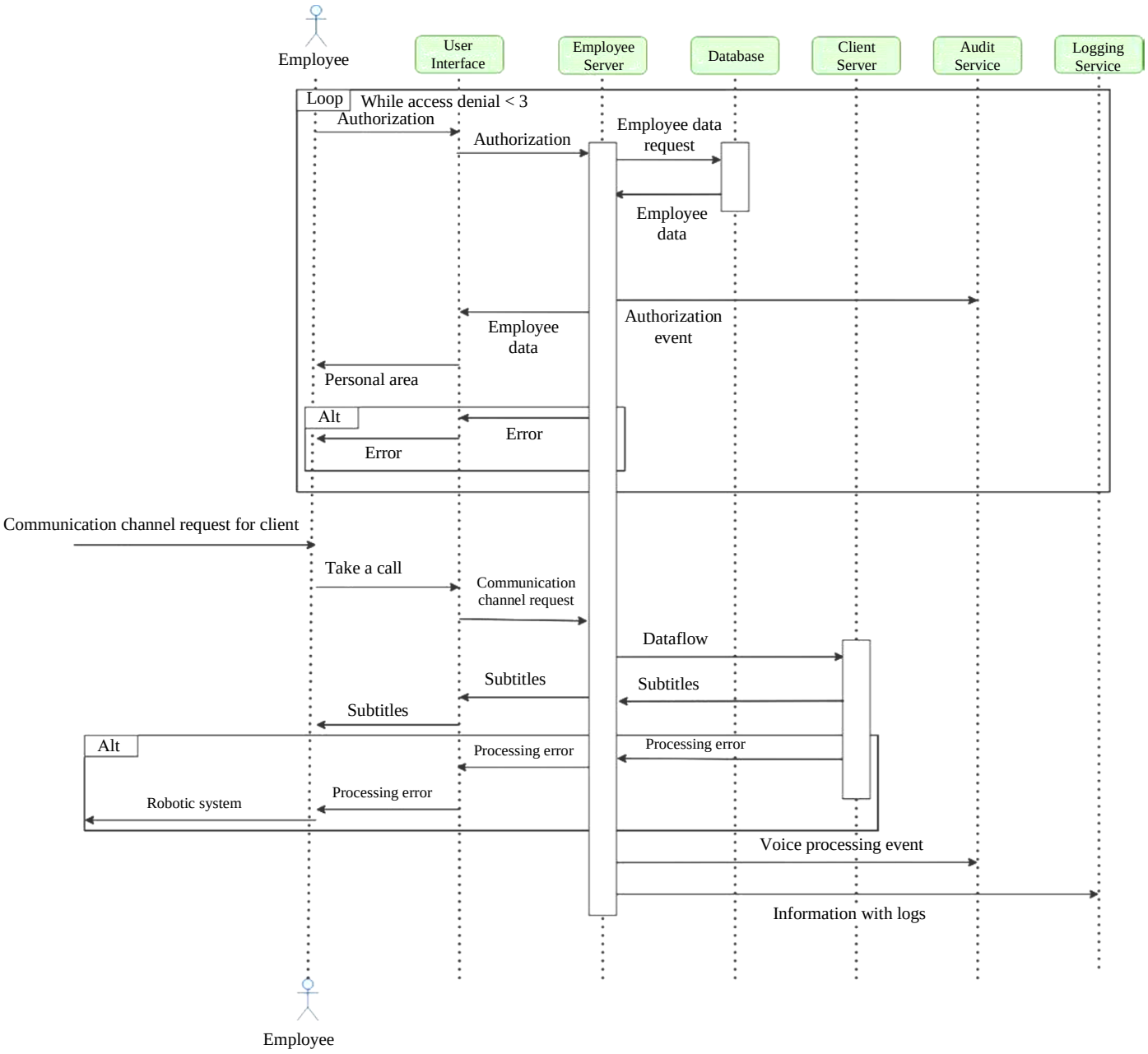

Fig. 2. Employee workflow diagram (the authors' figure)

igure 2 shows a workflow diagram from the point of view of an employee of the organization. After the standard authorization procedure, a call from the client may be received in the employee's personal account via a broadcast channel. If the call is accepted, a continuous information transmission channel is established between the client and the employee. The received data is transmitted in real time to the processing module, where text subtitles are generated using a set of transformations and methods.

In case of a conversion error, the client switches to a robotic system, and the employee is notified that voice processing is impossible. All actions (events) of the client and employee are marked in the speech recognition and logging module.

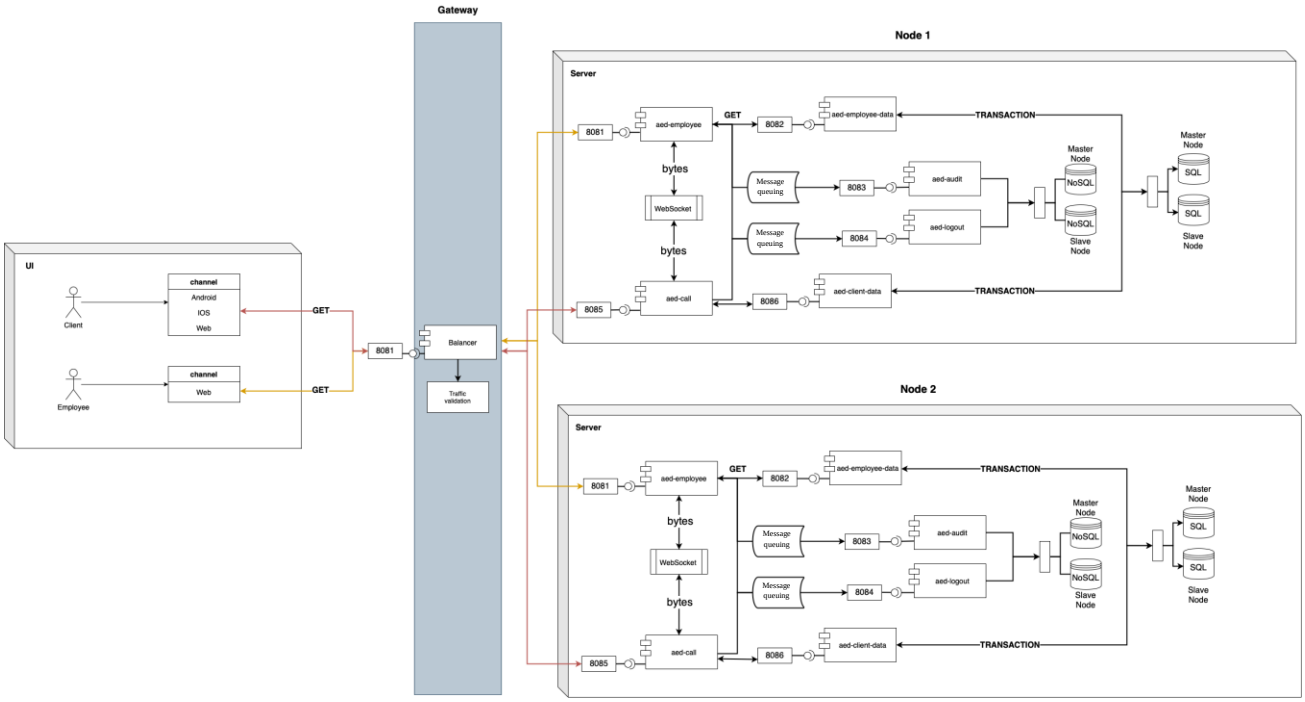

Fig. 3. System component deployment diagram (the authors' figure)

Figure 3 shows a microservice architecture project deployment diagram [4, 5], which consists of three blocks:

Internet. Visual modules for user-client interaction (User Interface, UI) are located on the Internet. It is worth noting that the client's channels for implementation are represented by three platforms: IOS, Android, Web; whereas the employee has only a Web channel. Principally, this is due to the fact that an employee does not usually need such applications for mobile operating systems because of the kind of work and security.

Data Gateway. The key task of the gateway is to provide the transfer of information from the Internet to the local network, where the primary servers are deployed, and back. This gives a number of advantages. First, there is a common point for transmitting all data. It is not possible to interact with the server bypassing the security gateway. Secondly, all requests can be checked for compliance with the originally laid format. Thirdly, the gateway allows load balancing between nodes, thereby achieving fault tolerance. If one node stops functioning, all requests are automatically distributed among other nodes. Fourthly, the system can be easily scaled. If the number of requests grows over time, it is enough to add a new node and the performance will remain at the same level.

Nodes. It is a complex system consisting of services interacting with each other. The major ones include services for receiving requests from a client and an employee. Their task is to receive a request, process it and set up interaction with each other using special components — sockets that allow data exchange in real time. These components receive data about the client and employee from special services that interact with a relational database using the SQL language [6]. For fault tolerance and speed, the database also has several nodes, but at least two. All user and client actions are transmitted to the speech recognition and logging services using message queues. This gives some advantages — asynchronous data transmission and safety of all information in case of a service failure. At that, storing such information in a relational database is impractical due to the uniformity and simplicity of the structure. For this reason, it is advisable to use a document-oriented NoSQL DBMS [7, 8][3]. It is worth noting that independent databases in each of the nodes provide high performance, and the consistency and relevance of data is implemented through a mechanism, whose replication will be described in detail in the following studies. The "Master-Slave" mechanism in

[3] Popov VB, Gavrikov IV. Tekhnologii "NOSQL" v algoritmakh analiza bol'shikh dannykh i iskusstvennom intellekte. In: "Problemy informatsionnoi bezopasnosti", Coll. of Papers V All-Russian Sci.-Pract. Conf. with international participation. Simferopol, Gurzuf, 2019. P. 158–160. (In Russ.)

relational and non-relational databases is required for possible vertical scaling and minimizing resource costs in case of an increase in the load on a single node.

All requests from the Internet to the local network occur over the HTTP protocol using the following methods:

– GET — in case of receiving information from any of the modules;

– POST — in case of creating records in any of the modules;

– PUT — in case of a change in any of the modules.

Requests to the stored information occur within transactions to provide data integrity [9–11].

Discussion and Conclusions. Architecture development, using the language of graphical description, enables to visually describe the functional requirements for the system, evaluate its effectiveness and reduce the number of errors.

The use case diagram made it possible to identify the need to develop a robotic system, as well as to identify many service options for a fuzzy model in case of exception situations during voice transcription. The deployment diagram helped to determine the effectiveness of the functions performed by each internal service, and the sequence diagram helped to determine the product lifecycle.

References

1. Baskakov AA, Tarasov AG. To the Problem of Using an Automated Workplace by People with Disabilities. Advanced Engineering Research. 2021;21:290–296. https://doi.org/10.23947/2687-1653-2021-21-3-290-296

2. Bazydto G, Adamski M, Węgrzyn M, et al. From UML State Machine Diagram into FPGA Implementation. IFAC Proceedings Volumes. 2013;46:298–303. https://doi.org/10.3182/20130925-3-CZ-3023.00061

3. Shailaja Uke, Ravindra Thool. UML Based Modeling for Data Aggregation in Secured Wireless Sensor Network. Procedia Computer Science. 2016;78:706–713. https://doi.org/10.1016/j.procs.2016.02.120

4. Saulo S de Toledo, Antonio Martini, Dag IK Sjøberg. Identifying Architectural Technical Debt, Principal, and Interest in Microservices: A Multiple-Case Study. Journal of Systems and Software. 2021;177:110968. https://doi.org/10.1016/j.jss.2021.110968

5. Nuno Mateus-Coelho, Manuela Cruz-Cunha, Luis Gonzaga Ferreira. Security in Microservices Architectures. Procedia Computer Science. 2021;181:1225–1236. https://doi.org/10.1016/j.procs.2021.01.320

6. Grigor'ev YuA, Matyukhin VG. Otsenka vremeni vypolneniya slozhnogo SQL-zaprosa v SUBD MS SQL SERVER 2000. Informatika i sistemy upravleniya. 2004;2:3–13. (In Russ.)

7. Davydov DA, Manylov PS. Evolution of NOSQL. Scientific and Technical Volga Region Bulletin. 2013;3:131–135.

8. Sharipova NN. About the Use of NOSQL Databases. Wschodnioeuropejskie czasopismo naukowe. 2016;9:73–76.

9. Chodak G, Suchacka G, Chawla Y. HTTP-Level E-Commerce Data Based on Server Access Logs for an Online Store. Computer Networks. 2020;183:107589. https://doi.org/10.1016/j.comnet.2020.107589

10. Okumura N, Ogata K, Shinoda Y. Formal Analysis of RFC 8120 Authentication Protocol for HTTP under Different Assumptions. Journal of Information Security and Applications. 2020;53:102529. https://doi.org/10.1016/j.jisa.2020.102529

11. Robert LR Mattson, Somnath Ghosh. HTTP-MPLEX: An Enhanced Hypertext Transfer Protocol and Its Performance Evaluation. Journal of Network and Computer Applications. 2009;32:925–939. https://doi.org/10.1016/j.jnca.2008.10.001

Received 10.09.2022.
Revised 24.10.2022.
Accepted 24.10.2022.

About the Authors:

Baskakov, Alexey A., post-graduate student of the Information Technologies and Computing Systems Department, Moscow State University of Technology (MSUT “STANKIN”) (1, Vadkovsky Lane, Moscow, 127055, RF), ORCID, aleks.baskakov@mail.ru

Tarasov, Alexey G., associate professor of the Information Technologies and Computing Systems Department, Moscow State University of Technology (MSUT “STANKIN”) (1, Vadkovsky Lane, Moscow, 127055, RF), Cand.Sci. (Eng.), ORCID, tarasov.ag@mail.ru

Claimed contributorship:

A. A. Baskakov: computational analysis; text preparation; search for scientific literature; formulation of conclusions. A. G. Tarasov: academic advising; research objectives and tasks setting; correction of the conclusions.

Conflict of interest statement

The authors do not have any conflict of interest.

All authors have read and approved the final manuscript.